Amendments to the Claims:

This listing of claims will replace all prior versions, and listing, of claims in the application:

Listing of Claims:

1. (Original) A system for transporting objects between a first and second machine where said first machine is programmed in a first language and said second machine is programmed in a second language, said system comprising:

a memory for storing code;

a first processor on said first machine for executing said code and instantiating an object on said first machine;

an output for outputting said object with persistence information to said second machine;

wherein, after said object is output from said first machine, said first processor deletes the instantiation of said object from said first machine.

2. (Original) The system according to claim 1, further comprising:

a second processor on said second machine for receiving said object with persistence information and allowing interaction with said object, said interaction creating events.

3. (Previously presented) The system according to claim 2, further comprising;

an output of said second machine for outputting said events and said objects with said persistence information to said first machine,

wherein said first machine reinstantiates said objects based on said persistence information and handles said events as effecting said reinstantiated objects.

4. (Original) A system for manipulating objects received at a first machine from a second machine, comprising:

an input in said first machine for receiving persistence information and an event from said second machine;

a processor in said first machine for instantiating an object based in part on said persistence information;

an event handler in said first machine for handling said event in combination with modifying said object;

an output for outputting said modified object to said second machine.

5. (Previously presented) A data structure for allowing the interchange of objects between a server and a client comprising:

a first object representation associated with an object;

persistence information associated with said first object representation;

event information relating to interaction with said object,

said object being instantiated on a first machine and being output from the first machine to a second machine, wherein after said object is output from said first machine, the instantiation of said object is deleted from the first machine.

6. (Original) A method for transporting objects between a first and second machine where said first machine is programmed in a first language and said second machine is programmed in a second language, said method comprising the steps of:

storing a code in a memory;

executing said code in a first processor on said first machine;

instantiating an object on said first machine;

outputting said object with persistence information to said second machine;

deleting said object from said first machine after said object is output from said first machine.

7. (Original) The method of according to claim 6, further comprising the steps of:

receiving said object with persistence information at a second processor on said second machine and

interacting with said object, said interaction creating events.

8. (Original) The method according to claim 7, further comprising the steps of:

outputting said events and said objects with said persistence information to said first machine;

reinstantiating said objects based on said persistence information; and

handling said events as effecting said reinstantiated objects.

9. (Original) A method for manipulating objects received at a first machine from a second machine, comprising the steps of:

receiving at a first machine persistence information and an event from said second machine;

instantiating an object based in part on said persistence information ~~in said first machine~~;

handling said event in combination with modifying said object;

outputting said modified object to said second machine.

10. (Previously presented) A system for transporting objects between a first and second machine where said first machine is programmed in a first language and said second machine is programmed in a second language, said system comprising:

a memory for storing code;

a first processor on said first machine for executing said code and instantiating an object on said first machine, said first machine being programmed in the first language;

an output of said first machine for transporting said object with persistence information to said second machine, said persistence information describing a property of said object and said second machine being programmed in the second language, wherein, after said object is transported from said first machine to said second machine, said first processor deletes the instantiation of said object from said first machine;

a second processor on said second machine for receiving said persistence information and instantiating said object received from said first machine on said second machine;

an output of said second machine for outputting said persistence information to said first machine,

wherein said first machine reinstantiates said object based on said persistence information.

11. (Previously presented) The system according to claim 10, further comprising:

a second processor on said second machine for receiving said object with persistence information and allowing interaction with said object, said interaction creating an event.

12. (Previously presented) The system according to claim 11, wherein said first machine handles said event as effecting the reinstantiated object on said first machine.

13. (Previously presented) A system for manipulating an object received at a first machine from a second machine, comprising:

an input in said first machine for receiving persistence information and an event from said second machine, said persistence information describing a property of said object;

a first processor in said first machine for instantiating said object on said first machine based in part on said persistence information;

an event handler in said first machine for handling said event in combination with modifying said object, said object being modified in said first machine;

an output on said first machine for outputting said modified object from said first machine to said second machine;

a second processor on said second machine for receiving and instantiating said modified object on said second machine.

14. (Previously presented) A data structure for allowing the interchange of objects between a server and a client comprising:

a first object representation associated with an object;

persistence information associated with said first object representation, said persistence information describing a property of said object;

event information relating to interaction with said object,

said object being instantiated on a first machine and being output from the first machine to a second machine, wherein after said object is output from said first machine, said object is instantiated on said second machine and the instantiation of said object in said first machine is deleted from said first machine and wherein after said object is instantiated on said second machine, an event is output with said persistence information from said second machine to the first machine and said object is reinstantiated on said first machine.

15. (Previously presented) A method for transporting objects between a first and second machine where said first machine is programmed in a first language and said second machine is programmed in a second language, said method comprising the steps of:

storing a code in a memory;

executing said code in a first processor on said first machine, said first machine being programmed in the first language;

instantiating an object on said first machine;

transporting said object with persistence information from said first machine to said second machine, said persistence information describing a property of said object and said second machine being programmed in the second language;

deleting said object from said first machine after said object is output from said first machine;

instantiating said object received with persistence information from said first machine on said second machine;

outputting said persistence information from said second machine to said first machine;

reinstantiating said object on said first machine based on said persistence information.

16. (Previously presented) The method of according to claim 15, further comprising the steps of:

receiving said object with persistence information at a second processor on said second machine and

interacting with said object, the interaction creating an event.

17. (Previously presented) The method according to claim 16, further comprising the step of:

handling said event as effecting the reinstantiated object.

18. (Previously presented) A method for manipulating an object received at a first machine from a second machine, comprising the steps of:

receiving at a first machine persistence information and an event from said second machine, said persistence information describing a property of said object;

instantiating said object based in part on said persistence information in said first machine;

handling said event in combination with modifying said object, said object being modified in said first machine;

outputting said modified object from said first machine to said second machine.

19. (New) A system for transporting objects between a first and second machine where said first machine is programmed in a first language and said second machine is programmed in a second language, said system comprising:

a memory for storing code and a page, said page comprising objects;

a first processor on said first machine for executing said code and instantiating at least one of said objects from said memory onto said first machine to create a first instantiated object on said first machine, said first instantiated object containing stored property information;

an output on said first machine for outputting said first instantiated object on said first machine from said first machine with persistence information to said second machine so that the first instantiated object is instantiated with persistence information on said second machine, said persistence information describing a property of said first instantiated object;

wherein, after said first instantiated object on said first machine is output from said first machine to said second machine, said first processor deletes the instantiation of said first instantiated object on said first machine from said first machine.

20. (New) The system according to claim 19, further comprising:

a second processor on said second machine for receiving said first instantiated object from said first machine after said first instantiated object on said first machine is output from said first machine to said second machine with persistence information and allowing interaction with said first instantiated object, said interaction creating an event.

21. (New) The system according to claim 20, further comprising;

an output of said second machine for outputting said event and said persistence information to said first machine, wherein said persistence information identifies said first instantiated object,

wherein said first machine receives said event and said persistence information from said second machine and reinstantiates said at least one of said objects onto said first machine based on said persistence information to create a reinstantiated object on said first machine corresponding to said first instantiated object and handles said event so that said reinstantiated object on said first machine is modified on said first machine based on said event to create a modified instantiated object on said first machine.

22. (New) A system for manipulating objects received at a first machine from a second machine, comprising:

an input in said first machine for receiving persistence information and an event from said second machine, said persistence information identifying an object instantiated on said second machine;

a processor in said first machine for instantiating an object based in part on said persistence information to create an instantiated object on said first machine, said instantiated object on said first machine corresponding to said object instantiated on said second machine;

an event handler in said first machine for handling said event received from said second machine in combination with modifying said instantiated object on said first machine wherein said event causes a modification in a property of said instantiated object on said first machine to create a modified object instantiated on said first machine;

an output for outputting said modified object instantiated on said first machine to said second machine with persistence information describing said modified object instantiated on said first machine.

23. (New) A data structure for allowing the interchange of objects between a server and a client comprising:

a first object representation associated with an object;

persistence information associated with said first object representation;

event information relating to interaction with said object,

said object being instantiated on a first machine and being deleted from said first machine after the object is output from the first machine to a second machine, wherein the second machine receives said object from said first machine with persistence information, said persistence information describing a property of said object, and wherein the second machine outputs said persistence information with an event to said first machine, said event causing a modification of the object on said first machine.

24. (New) A method for transporting objects between a first and second machine where said first machine is programmed in a first language and said second machine is programmed in a second language, said method comprising the steps of:

storing a code in a memory;

executing said code in a first processor on said first machine;

instantiating an object on said first machine, said object containing state information;

outputting said object from said first machine with persistence information to said second machine, said persistence information describing a property of said object;

deleting said object from said first machine after said object is output from said first machine to said second machine.

25. (New) The method of according to claim 24, further comprising the steps of:

receiving said object with persistence information at a second processor on said second machine and

interacting with said object, said interaction creating an event, said event describing a modification of said object.

26. (New) The method according to claim 25, further comprising the steps of:

outputting said event and said persistence information to said first machine from said second machine;

reinstantiating said object based on said persistence information received from said second machine; and

handling said event as effecting said reinstantiated objects to create a modified object on said first machine.

27. (New) A method for manipulating objects received at a first machine from a second machine, comprising the steps of:

receiving at a first machine persistence information and an event from said second machine, said persistence information describing a property of an object instantiated on said second machine;

instantiating an object in said first machine based in part on said persistence information received from said second machine to create an instantiated object on said first machine, said instantiated object on said first machine corresponding to said object instantiated on said second machine;

handling said event in combination with modifying said instantiated object on said first machine to create a modified object on said first machine;

outputting said modified object from said first machine to said second machine.

28. (New) A system for transporting objects between a first and second machine where said first machine is programmed in a first language and said second machine is programmed in a second language, said system comprising:

a memory for storing code;

a first processor on said first machine for executing said code and instantiating an object on said first machine from said memory to create a first instantiated object on said first machine, said first machine being programmed in the first language;

an output of said first machine for transporting said first instantiated object with persistence information to said second machine so that the first instantiated object is capable of being instantiated on said second machine with said persistence information, said persistence information describing a property of said first instantiated object and said second machine being programmed in the second language, wherein, after said first instantiated object is transported

from said first machine to said second machine, said first processor deletes the instantiation of said first instantiated object from said first machine;

a second processor on said second machine for receiving said persistence information associated with said first instantiated object and instantiating said first instantiated object received from said first machine on said second machine;

an output of said second machine for outputting said persistence information to said first machine,

wherein said first machine reinstantiates said first instantiated object from said memory based on said persistence information.

29. (New) The system according to claim 28, further comprising:

a second processor on said second machine for receiving said first instantiated object from said first machine with persistence information so that said first instantiated object is capable of being instantiated on said second machine and allowing interaction with said first instantiated object on said second machine, said interaction creating an event at said second machine.

30. (New) The system according to claim 29, wherein after said first machine reinstantiates said first instantiated object from said memory based on said persistence information received from said second machine, said first machine handles said event as effecting the reinstantiated object on said first machine to create a modified instantiated object on said first machine.

31. (New) A system for manipulating an object received at a first machine, comprising:

an input in said first machine for receiving persistence information and an event from a second machine, said persistence information describing a property of said object, said object instantiated on said second machine;

a first processor in said first machine for instantiating an object corresponding to said object instantiated on said second machine on said first machine based in part on said persistence information received from said second machine to create an instantiated object on said first machine corresponding to said object instantiated on said second machine;

an event handler in said first machine for handling said event received from said second machine in combination with modifying said instantiated object on said first machine to create a modified object instantiated on said first machine, said modified object instantiated on said first machine being modified from said instantiated object on said first machine;

an output on said first machine for outputting said modified object from said first machine to said second machine;

a second processor on said second machine for receiving and instantiating said modified object on said second machine.

32. (New) A data structure for allowing the interchange of objects between a server and a client comprising:

a first object representation associated with an object;

persistence information associated with said first object representation, said persistence information describing a property of said object;

event information relating to interaction with said object,

said object being instantiated on a first machine and being output from the first machine to a second machine with said persistence information, wherein after said object is output from said first machine, said object is instantiated on said second machine and the instantiation of said object in said first machine is deleted from said first machine and wherein after said object is instantiated on said second machine with said persistence information, an event is output with said persistence information from said second machine to the first machine and said object is reinstantiated on said first machine based on said persistence information received from said second machine.

33. (New) A method for transporting objects between a first and second machine where said first machine is programmed in a first language and said second machine is programmed in a second language, said method comprising the steps of:

storing a code in a memory;

executing said code in a first processor on said first machine, said first machine being programmed in the first language;

instantiating an object on said first machine;

transporting said object with persistence information from said first machine to said second machine, said persistence information describing a property of said object and said second machine being programmed in the second language;

deleting said object from said first machine after said object is output from said first machine;

instantiating said object received with persistence information from said first machine on said second machine;

outputting said persistence information from said second machine to said first machine;

receiving said persistence information at said first machine from said second machine and reinstantiating said object on said first machine based on said persistence information received at said first machine from said second machine.

34. (New) The method of according to claim 33, further comprising the steps of:

receiving said object with persistence information at a second processor on said second machine from said first machine and

interacting with said object at said second machine, the interaction creating an event at said second machine.

35. (New) The method according to claim 34, further comprising the steps of:

transporting said event from said second machine to said first machine;

handling said event at said first machine after receiving said persistence information from said second machine at said first machine and after reinstantiating said object on said first machine based on said persistence information received from said second machine to create a reinstantiated object on said first machine, said event being received at said first machine from said second machine and effecting the reinstantiated object to create a modified object on said first machine.

36. (New) A method for manipulating an object received at a first machine, comprising the steps of:

receiving at a first machine persistence information and an event from a second machine, said persistence information describing a property of said object, said object being instantiated on said second machine;

instantiating said object based in part on said persistence information received from said second machine in said first machine;

handling said event in combination with modifying said object, said object being modified in said first machine to create a modified object in said first machine;

outputting said modified object from said first machine to said second machine with modified persistence information, said modified persistence information corresponding to a modified property of said modified object.

37. (New) The system according to claim 21 wherein said first machine outputs said modified instantiated object from said first machine to said second machine with modified persistence information, said modified persistence information corresponding to a modified property of the first instantiated object.